REMARKS

Claims 1-25 are pending, of which claims 1 and 8 are independent method claims with generally corresponding computer program product claims 13 and 20, and claim 25 is an independent system.

The Office Action rejected claims 1-25 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,061,740 to Ferguson at al. ("*Ferguson*") in view of U.S. Patent No. 6,385,618 to Ng et al. ("*Ng*") and U.S. Patent No. 6,021,331 to Cooper et al. ("*Cooper*").[1]

Applicants' invention, as claimed for example in independent method claim 1 relates to supporting different security descriptor specifications for the same object. The method includes converting a first security descriptor into a version of the first security descriptor that follows a second security descriptor specification, comparing the converted version of the first security descriptor with a second security descriptor, and changing the second security descriptor to reflect at least one security permission change as represented in the converted version of the first security descriptor so that any changes to the second security descriptor are non-degenerative and reversible. The method further includes undoing the at least one security permission change in the second security descriptor, converting the second security descriptor into a version of the second security descriptor that follows the first security descriptor specification, comparing the converted version of the second security descriptor with the first security descriptor, and changing the first security descriptor to reflect the undone security permission change as represented in the converted version of the second security descriptor so that any change to the first security descriptor is non-degenerative and reversible. Independent claims 13 and 25 recite similar limitations from the perspective of a computer program product and computer system, respectively.

Applicants' invention, as claimed for example in independent method claim 8 relates to replicating in a non-degenerative fashion a first security descriptor with a second security descriptor specification. The method includes consulting mapping rules that define mappings of rights between the first security descriptor specification and the second security descriptor specification and for each right of the first security descriptor specification for which there is a

[1]Although the prior art status of *Ferguson*, *Ng*, and *Cooper* is not being challenged at this time, Applicants reserve the right to do so in the future. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status or asserted teachings of *Ferguson*, *Ng*, and *Cooper*.

corresponding mapping rule, converting the right that follows the first security descriptor specification to a corresponding right that follows the second security descriptor specification. The method assembles each converted right that follows the second security descriptor specification to form a version of the first security descriptor that follows the second security descriptor specification and compares each converted right in the version of the first security descriptor that follows the second security descriptor specification to the corresponding right in the second security descriptor. Based on the comparing of each converted right in the version of the first security descriptor that follows the second security descriptor specification, the method detects one or more changes in the converted first security descriptor that are not reflected in the second security descriptor and changes the second security descriptor to reflect the detected one or more changes in the first security descriptor so that changes to the second security descriptor are non-degenerative and reversible.

The method further includes changing one or more rights in the second security descriptor and for each right of the second security descriptor specification for which there is a corresponding mapping rule, converting the right that follows the second security descriptor specification to a corresponding right that follows the first security descriptor specification. The method assembles each converted right that follows the first security descriptor specification to form a version of the second security descriptor that follows the first security descriptor specification and compares each converted right in the version of the second security descriptor that follows the first security descriptor specification to the corresponding right in the first security descriptor. Based on the comparing of each converted right in the version of the second security descriptor that follows the first security descriptor specification, the method detects one or more changes in the converted second security descriptor that are not reflected in the first security descriptor and changes the first security descriptor to reflect the detected one or more changes in the second security descriptor so that changes to the first security descriptor are non-degenerative and reversible. Independent claim 20 recites similar limitations from the perspective of a computer program product.

In order to establish a *prima facie* case of obviousness, "the prior art reference (or references when combined) must teach or suggest all the claim limitations." MPEP § 2143 (emphasis added). During examination, the pending claims are given their broadest reasonable interpretation, i.e., they are interpreted as broadly as their terms reasonably allow, consistent with

the specification. MPEP §§ 2111 & 2111.01. Applicants respectfully submit, however, that for at least the reasons stated below *Ferguson*, *Ng*, and *Cooper* fail to teach or suggest all the claim limitations of independent claims 1, 8, 13, 20, and 25.

Specifically, *Ferguson*, *Ng*, and *Cooper* fail to teach or suggest the non-degenerative and reversible features of Applicants' invention with respect to different security descriptors that follow first and second security descriptor specifications. For example, among other things, *Ferguson*, *Ng*, and *Cooper* fail to teach or suggest changing a second security descriptor to reflect at least one security permission change as represented in a converted version of a first security descriptor, undoing the at least one security permission change in the second security descriptor, and changing the first security descriptor to reflect the undone permission change as represented in a converted version of the second security descriptor, as recited in claims 1, 13, and 25, and fail to teach or suggest changing a second security descriptor to reflect one or more changes detected in a first security descriptor, changing one or more rights in the second security descriptor, and changing the first security descriptor to reflect one or more changes detected in the second security descriptor, as recited in independent claims 8 and 20.

As noted in Applicants' prior response, *Ferguson* discloses an administration system for centralized management of a heterogenous network. Col. 8, ll. 41-42. With reference to Figure 4, a management service includes a set of representation objects 90 within a distributed directory for representing foreign objects 87. Col. 9, ll. 1-3. When a change to representation object 90 is detected, an event monitor sends a message to a replication agent 89 to synchronize foreign objects 87. Col. 9, ll. 8-32. The administration system acts as a one-way synchronization between replication objects 90 and the foreign objects 87. Col. 9, ll. 35-37. In an example illustrated in Figure 6 that involves Microsoft's Security Accounts Manager ("SAM") and Novell Directory Services ("NDS"), *Ferguson* states that NDS values trump SAM values. Col. 14, ll. 33-35. Accordingly, based on the example illustrated in Figure 6, *Ferguson's* changes are not non-degenerative and reversible.

Applicants' prior response also noted that *Ng* discloses an object-relational mapping tool. Col. 3, ll. 33-38. The object-relational mapping tool reads a database to examine its schema, constructs a data structure to reflect this schema, generates an object model based on the data structure, and creates source code based on the object model. Col. 5, ll. 23-27. Using the object-relational mapping tool, a programmer can customize the object model. Col. 6, ll. 4-5.

After the programmer customizes the object model, however, a database administrator may update the database. Col. 7, ll. 5-12. In order to keep from losing the customizations when the programmer updates the source code to include the database update, the object relational mapping tool, imports the new database schema to create a new data structure, compares the old data structure with the new data structure to isolate the database changes, updates the object model to reflect the identified database changes without disturbing the changes made by the programmer, and generates new source code from the updated object model. Col. 7, ll. 13-60. Accordingly, similar to *Ferguson*, *Ng* also discloses a one-way operation from schema to data structure to object model to source code.

Cooper discloses a client station for controlling a telecommunications system. Col. 1, l. 62. Cooper indicates that all operations are reversible. Col 14, ll. 22. The reversible nature is provided by four data classes which provide an address of the destination object, a flag indicating whether the operation is invertible, a before state, and an after state. Col. 14, ll. 22-25. Upon arrival of an operation at a destination object and before application of new state parameters to the destination object, the current state parameters (i.e., the destination object's state prior to application of the operation) are stored in the before state. Col. 14, ll. 31-35. The parameters stored in the after state may then be applied to the destination object to effect a change. Col. 14, ll. 35-37. When the before state is filled, the invertible flag is implemented to indicate that the operation is now reversible. Col. 14, ll. 37-39. Beginning at line 53 of column 16, *Cooper* describes undo/redo in more detail.

Note, however, that *Cooper* merely describes making and undoing changes to a single destination object. Accordingly, even assuming for the sake of argument that it is proper to combine *Cooper* with *Ferguson* and *Ng*, the combination does not teach that changes made to NDS values could be undone through corresponding changes to SAM values (*see Ferguson*) or that changes made to a schema could be undone through corresponding changes to a data structure, changes made to a data structure could be undone through corresponding changes to an object model, changes made to an object model could be undone through corresponding changes to source code, etc. (*see Ng*). In other words, there is no indication that *Cooper's* saved parameters have any meaning in a context other than the object for which they were created.

Accordingly, *Ferguson*, *Ng*, and *Cooper* fail to teach, suggest, or motivate non-degenerative and reversible changes to different security descriptors for the same object.

Among other things, therefore, as noted above, *Ferguson*, *Ng* , and *Cooper* fail to teach, suggest, or motivate: changing a second security descriptor to reflect at least one security permission change as represented in a converted version of a first security descriptor, undoing the at least one security permission change in the second security descriptor, and changing the first security descriptor to reflect the undone permission change as represented in a converted version of the second security descriptor, as recited in claims 1, 13, and 25, and fail to teach or suggest changing a second security descriptor to reflect one or more changes detected in a first security descriptor, changing one or more rights in the second security descriptor, and changing the first security descriptor to reflect one or more changes detected in the second security descriptor, as recited in independent claims 8 and 20. Applicants respectfully submit, therefore, that the rejection of the pending claims under 35 U.S.C. § 103(a) as unpatentable over *Ferguson* in view of *Ng* and *Cooper* should be withdrawn.

With respect to Applicants' prior arguments regarding the asserted motivation to combine *Ferguson* and *Ng*, the Office Action states that "Ng teaches other beneficial results than found in Ferguson and the prior art reference must be considered in its entirety." Office Action, p. 2. Applicants acknowledge that it is appropriate to consider prior art references in their entirety. *See* MPEP 2141.02 ("A prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention."). However, in supporting the combination of *Ferguson* and *Ng*, the Office Action continues to assert a motivation that, as Applicants explained in their prior response, is contrary to their combination. Specifically, the Office Action asserts that "it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the method of *Ferguson* . . . in order to alleviate programmers from having to recreate their customization . . . as taught by *Ng* . . . to save significant development time." Office Action, p. 4 (rejection of claim 1).

As indicated previously, *Ferguson* teaches that NDS values trump SAM values, which is contrary to Applicants' claimed invention for non-degenerative and reversible changes, as claimed for example in independent claims 1, 8, 13, 20, and 25, and is contrary to *Ng's* goal of preserving programmer customizations to the object model. Keeping in mind that *Ferguson's* teaching that NDS values trump SAM values is in connection with an integration utility for user and group objects underscores the relevance of *Ferguson's* contrary position with respect to Applicants claimed invention. Accordingly, Applicants respectfully request that the Examiner

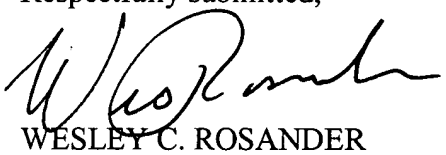
state the "other beneficial results" so that the record is clear with respect to the asserted motivation for combining *Ferguson* and *Ng* in order to provide Applicants a fair opportunity to rebut the asserted motivation. *See* MPEP § 2145(X)(D)(2) ("It is improper to combine references where the references teach away from their combination."); MPEP § 2145(X)(D)(1) ("A prior art reference that 'teaches away' from the claimed invention is a significant factor to be considered in determining obviousness."). Making the record clear with respect to the asserted motivation for combining *Ferguson* and *Ng* also will give Applicants an opportunity to evaluate whether the combination is based on impermissible hindsight reasoning. *See* MPEP § 2145(X)(A). For purposes of this response, Applicants maintain that the asserted motivation to combine *Ferguson* and *Ng* as stated in the Office Action is improper.

Based on at least the foregoing reasons, Applicants respectfully submit that the cited prior art fails to anticipate or make obvious Applicants invention, as claimed for example, in independent claims 1, 8, 13, 20, and 25. Applicants note for the record that the remarks above render the remaining rejections of record for the independent and dependent claims moot, and thus addressing individual rejections or assertion with respect to the teachings of the cited art is unnecessary at the present time, but may be undertaken in the future if necessary or desirable, and Applicants reserve the right to do so.

In the event that the Examiner finds any remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney.

Dated this 24th day of August, 2004.

Respectfully submitted,



WESLEY C. ROSANDER
Registration No. 51,030
ERIC M. KAMERATH
Registration No. 46,081
RICK D. NYDEGGER
Registration No. 28,651
Attorneys for Applicant
Customer No. 022913

EMK:kc
KC0000002864V001